

Pregătire CEX, 12 noiembrie 2011

Prof. Gabriela Conea, Colegiul National
gconea2002@yahoo.com

Limbajul C++

Tipuri structurate: tablouri unidimensionale (vectori)

Documentatie: <http://www.cplusplus.com/doc/tutorial/arrays/>

Prezentare

- Pornind de la **tipurile fundamentale** de date se pot obține tipuri mai complexe de date, același identificator de variabilă grupând mai multe elemente, fiecare element putând fi accesat în mod individual obținând astfel **un tip structurat**.
- Un tip structurat este **tabloul**, definit ca o **structură care cuprinde un număr de componente de același tip, memorate într-o zonă de memorie contiguă**.
- Tipul componentelor se numește tipul de bază al tabloului.
- Accesarea unei componente din cadrul tabloului se realizează prin intermediul unui **indice**, valorile indicilor identificând în mod unic componentele.
- În cazul în care componentele unui tablou sunt structurate după valorile unui singur indice tabloul se numește **unidimensional**, iar dacă ele sunt structurate după valorile mai multor indici tabloul se numește **multidimensional**, fiecare indice definind o dimensiune.

Declarare: tip numet[nrelem];

ex: int a [10]; declară tabloul a cu 10 elemente de tip întreg

| | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|
| | | | | | | | | | |
| a[0] | a[1] | a[2] | a[3] | a[4] | a[5] | a[6] | a[7] | a[8] | a[9] |

Obs: elementele tabloului pot fi inițialitate chiar de la declarare:

tip numet[nrelem]={val0, val1,..., valk};

ex: int a[]={12,11, 2011, 10};

| | | | |
|------|------|------|------|
| 12 | 11 | 2011 | 10 |
| a[0] | a[1] | a[2] | a[3] |

Acces la componente: numet[indice];

Ex: a[0], a[1], ..., a[9];

Prelucrări elementare:

a) Citirea unui tablou unidimensional:

```
float a[100];  
int n;  
cin>>n;  
for(int i=0;i<n;i++) cin>> a[i];
```

b) Afisarea unui tablou unidimensional:

```
for(int i=0;i<n;i++) cout<< a[i]<<' ';
```

Aplicatii

I.

1. Fie a și b doi vectori cu câte n , respectiv m elemente. Afișați toate elementele din vectorul a ce sunt strict mai mici decât fiecare element din vectorul b . Dacă nu există astfel de elemente precizați mesaj adecvat.
2. Fie un vector ce conține n elemente de tip `char`; creați un program ce afișează perechea de două elemente egal depărtate de centru a căror sumă a codurilor ASCII este maximă.

II.

1. Ordonarea crescătoare a elementele unui vector cu n elemente: metoda bubble sort.

2. Vector de frecvență:

1.

Se consideră fișierul **DATE.IN** care conține cel mult 100000 de numere naturale formate fiecare din cel mult opt cifre, separate prin câte un spațiu. Scrieți un program C sau C++ care scrie în fișierul **DATE.OUT** o valoare care reprezintă numărul de cifre care **NU** au apărut în niciunul din numerele aflate în fișierul **DATE.IN**.

Exemple:

| DATE.IN | DATE.OUT |
|--------------------------------|----------|
| 12 222 12 21 87 6 89 788 3 900 | 2 |

Explicații:

-sunt două cifre care nu au apărut în scrierea numerelor din fișierul **DATE.IN**. și anume 4 și 5

2.

Fișierul **BAC.TXT** conține pe prima linie un număr natural cu exact 2000 de cifre din mulțimea $\{0, 1, \dots, 9\}$. Cifrele numărului nu sunt separate prin spații.

a) Scrieți un program C/C++ care afișează pe ecran numărul cel mai mare care se poate obține din cifrele numărului citit din fișierul **BAC.TXT**. Se va utiliza un algoritm eficient din punct de vedere al timpului de executare.

Exemplu: dacă fișierul conține numărul 2417400...0 se va afișa 7442100...0
1995 de 0 1995 de 0

b) Descrieți metoda utilizată și explicați în ce constă eficiența ei.

III. Campion

1. problema: [cifre](#), [roci](#), schi